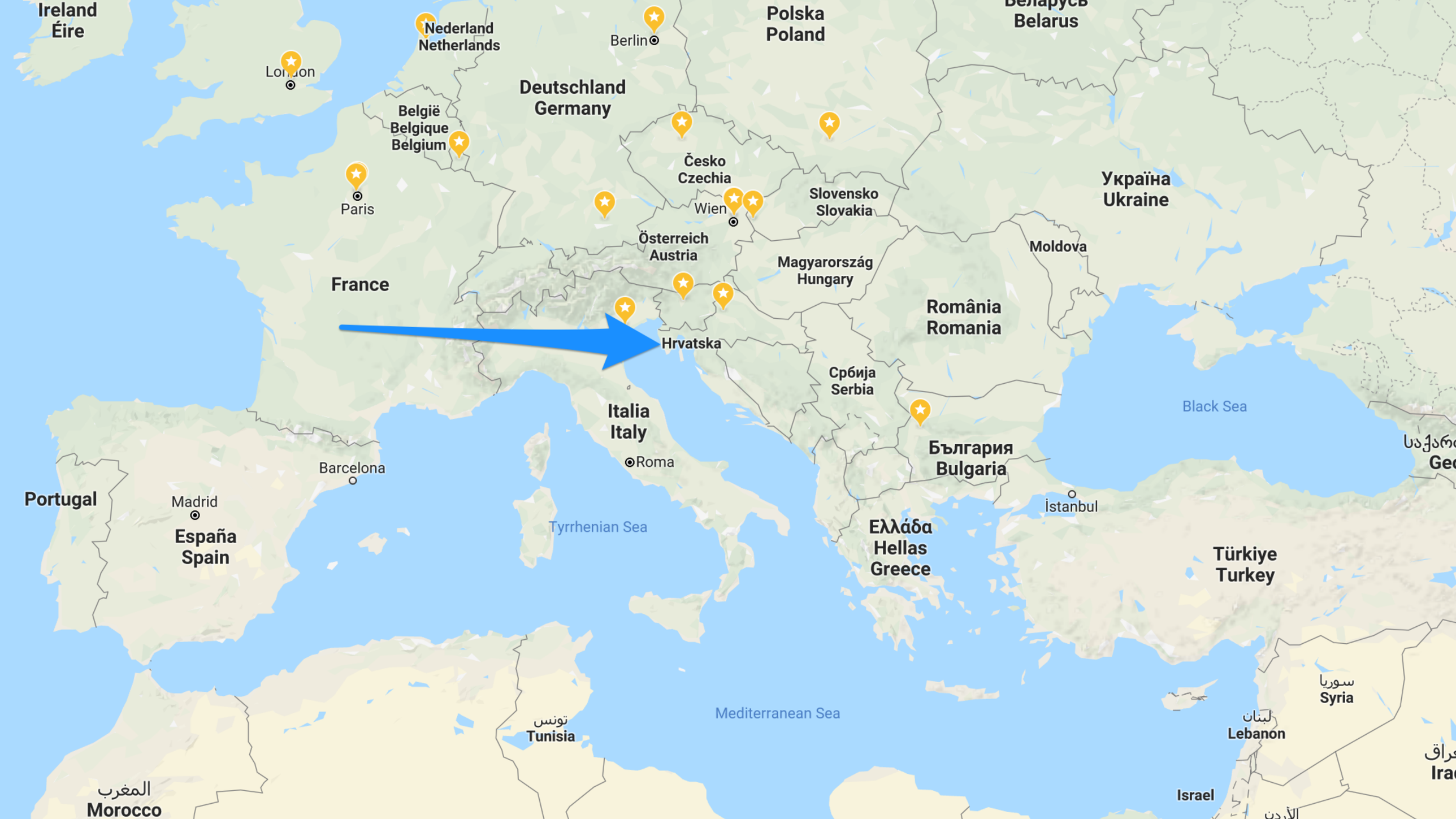
ANA BAOTIĆ

BREAK YOUR APP BEFORE SOMEONE ELSE DOES


Ireland
Éire
London
Nederland
Netherlands
Berlin
Polska
Poland
Беларусь
Belarus
Deutschland
Germany
België
Belgique
Belgium
Česko
Czechia
Slovensko
Slovakia
Україна
Ukraine
Paris
Wien
Österreich
Austria
Magyarország
Hungary
Moldova
France
România
Romania
Hrvatska
Србија
Serbia
Black Sea
Italia
Italy
Roma
България
Bulgaria
Barcelona
Portugal
Madrid
España
Spain
İstanbul
Tyrrhenian Sea
Ελλάδα
Hellas
Greece
Türkiye
Turkey
Mediterranean Sea
سوريا
Syria
لبنان
Lebanon
تونس
Tunisia
المغرب
Morocco
Israel

AGENDA

- Code coverage
- Android/APK
- Tools
- Conclusion

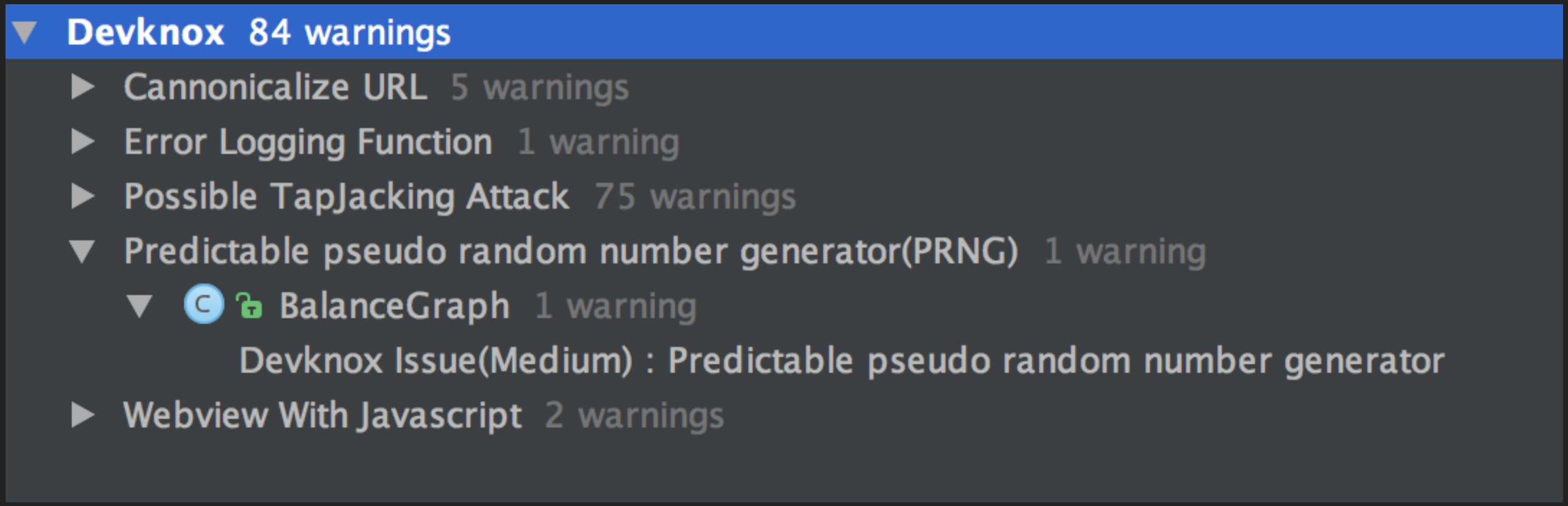
STATIC ANALYSIS

- pmd, findbugs, checkstyle
- lint http://tools.android.com/tips/lint-checks

```
SecureRandom
------------
Summary: Using a fixed seed with SecureRandom

Priority: 9 / 10
Severity: Warning
Category: Security

Specifying a fixed seed will cause the instance to return a predictable
sequence of numbers. This may be useful for testing but it is not
appropriate
for secure use.

More information:
http://developer.android.com/reference/java/security/SecureRandom.html
▸
```

DEVKNOX

- AS plugin (lite)
- tool for detecting security issues
- scan modules/whole app/selection


Devknox 84 warnings
Cannonicalize URL 5 warnings
Error Logging Function 1 warning
Possible TapJacking Attack 75 warnings
Predictable pseudo random number generator(PRNG) 1 warning
BalanceGraph 1 warning
Devknox Issue(Medium) : Predictable pseudo random number generator
Webview With Javascript 2 warnings

We are no longer supporting devknox offically.
An open source version will be released soon.

¯_(ツ)_/¯

AUTOMATED TESTING

- Espresso, Robotium, (Robolectric)
- UI, business logic

QA

- let someone else use the app
- real, physical devices!

GOOGLE PLAY (BETA/ALPHA)

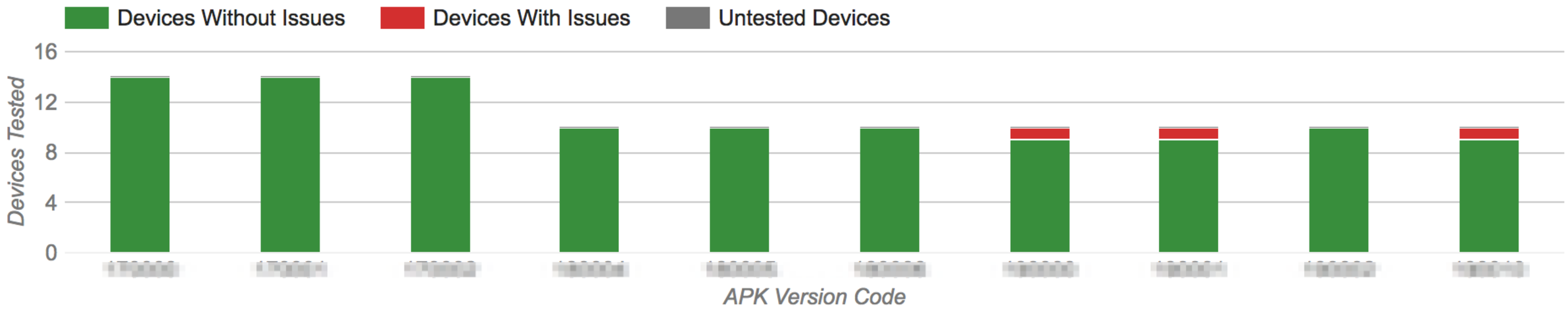
- let someone else use the app
- real, physical devices!

CRASHES | PERFORMANCE | SCREENSHOTS | SECURITY | SETTINGS

Crashes show issues found when the Firebase Test Lab physical devices launched and interacted with your app.

APK Launch Comparison

PRE-LAUNCH REPORT NOTIFICATIONS: ON

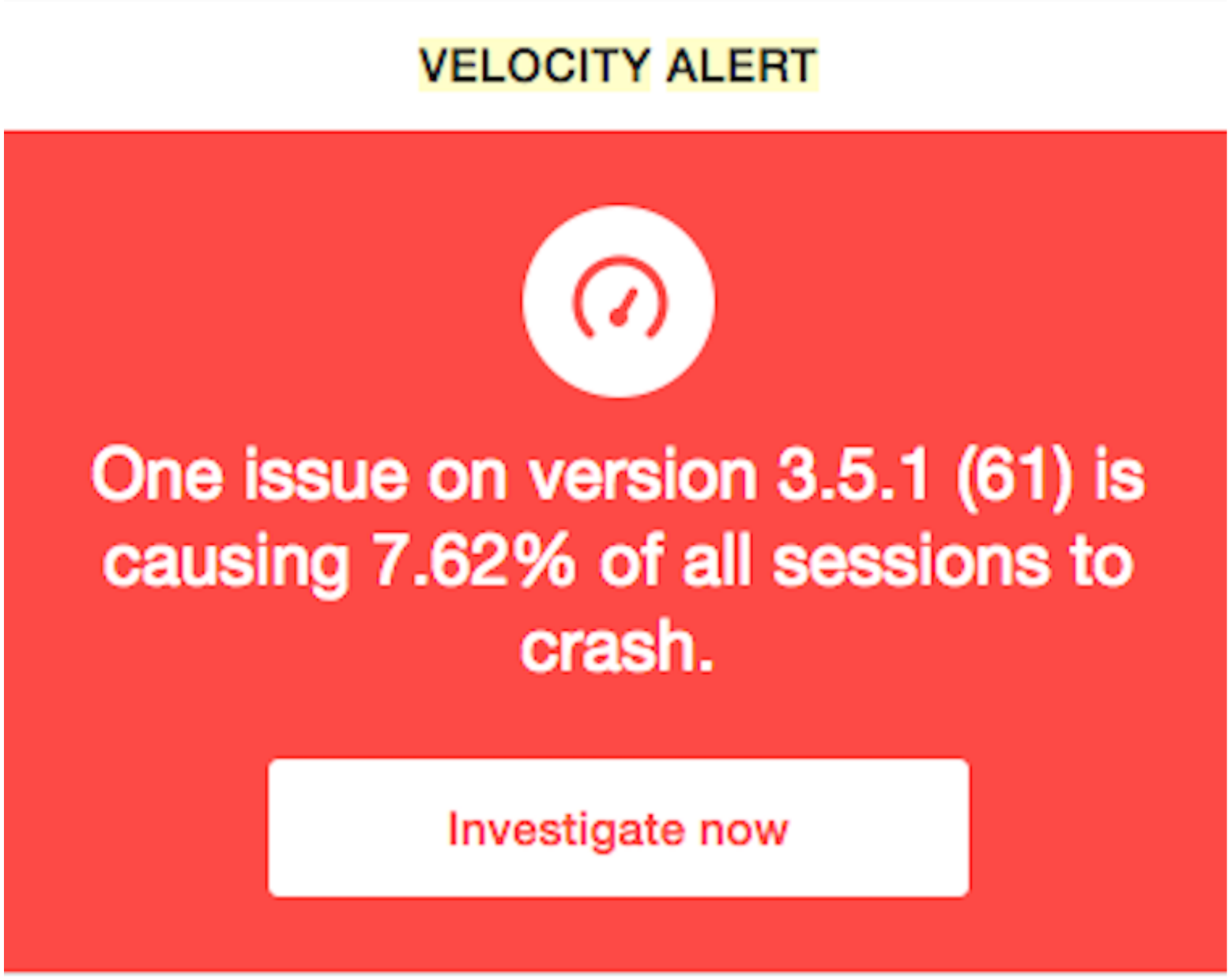


Test results for APK version [redacted]

ALL | ISSUES

Devices With Issues	Devices Without Issues	Devices Tested
0	14	14

AKA PRODUCTION



line 131

Crashes
219

Users
1

Overview

Stacktrace

Notes

Version 3.1.1 (57)

Operating system

100.0% 4.3

Devices

100.0% HUAWEI Y530-U00


Galaxy S4 (GT-I9505)
Nexus 4
Galaxy S II (GT-I9100)
Galaxy Nexus
Nexus 7
DROID RAZR
GT-I9500
HTC One X
GT-P3110
Desire HD
MID
C5170
EVO
ST23i
ST26a
CM980
U9200
MB88
E860
MI 1S
W732
LT18a
A1
A8
A501
Xoom

PENETRATION TESTING

*An **authorised**, **simulated** attack on a computer system that looks for security weaknesses, potentially gaining access to the system's features and data.*

MOTIVATION

- *security*
- *privacy*
- *mandated by industry*

GET THE APK


WHAT IF I TOLD YOU
YOU CAN LOOK AT OTHER APPS?

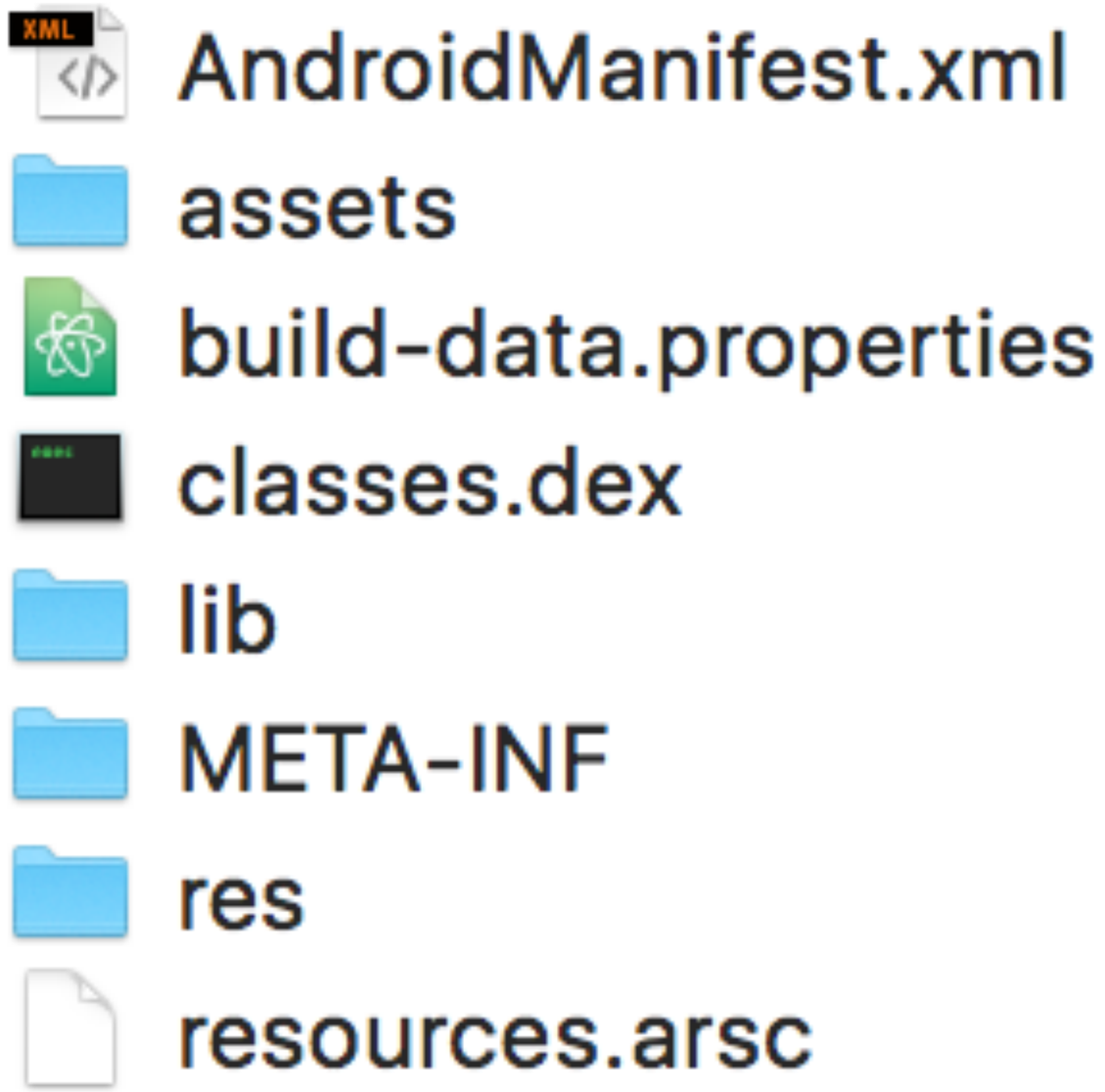
```
➜ ~ adb shell pm list packages

package:com.mobeam.barcodeService
…
package:com.sec.android.widgetapp.samsungapps
package:com.google.android.youtube
package:com.samsung.android.app.galaxyfinder
package:com.samsung.android.themestore
package:com.sec.android.app.chromecustomizations
package:com.samsung.android.videolist
package:com.samsung.android.video
…
package:com.samsung.android.videolist
package:com.samsung.android.video
```

```
➜ ~ adb shell pm list packages | grep "samsung"

…
package:com.samsung.android.coreapps
package:com.samsung.android.videolist
package:com.samsung.android.video
package:com.samsung.android.videolist
package:com.samsung.android.video
```

```
➜  ~ adb shell pm path com.samsung.android.video

package:/system/priv-app/SamsungVideoPlayer_DreamPreview/
SamsungVideoPlayer_DreamPreview.apk
```

```
➜ ~ adb pull /system/priv-app/
SamsungVideoPlayer_DreamPreview/
SamsungVideoPlayer_DreamPreview.apk .


/system/priv-app/SamsungVideoPlayer_DreamPreview/
SamsungVideoPlayer_DreamPreview.apk: 1 file pulled.
14.3 MB/s (3866839 bytes in 0.257s)
```

APK – ANDROID PACKAGE KIT



TOOLS

- aapt
- apktool
- classychark
- androguard
- Charles

AAPT – ANDROID ASSET PACKAGING TOOL

- in build tools
- insight into resources and apk

```
➜ aapt list cool.apk

AndroidManifest.xml
META-INF/*
assets/become_user_en.html
assets/location_default.json
assets/style.css
res/anim/*
res/drawable*
res/layout/*
res/menu/*
res/raw/*
res/xml/*
resources.arsc
```

```
➜ aapt dump strings cool.apk

String pool of 5568 unique UTF-8 non-sorted strings, 5568
entries and 0 styles using 262112 bytes:
String #0: res/menu/sort_menu.xml
String #1: res/color/
abc_btn_colored_borderless_text_material.xml
String #2: res/drawable/abc_btn_borderless_material.xml
String #3: res/drawable/abc_btn_check_material.xml
…
```

```
➜ aapt dump xmlstrings cool.apk AndroidManifest.xml


String pool of 168 unique UTF-16 non-sorted strings,
168 entries and 0 styles using 12972 bytes:
String #0: installLocation
String #1: versionCode
String #2: versionName
String #3: minSdkVersion
String #4: targetSdkVersion
String #5: name
String #6: protectionLevel
…
```

```
➜ aapt dump permissions cool.apk

package: abaotic.demo.development
uses-permission: name='android.permission.ACCESS_FINE_LOCATION'
uses-permission: name='android.permission.ACCESS_COARSE_LOCATION'
uses-permission: name='android.permission.INTERNET'
uses-permission: name='android.permission.CALL_PHONE'
uses-permission: name='android.permission.CAMERA'
uses-permission: name='android.permission.READ_PHONE_STATE'
uses-permission: name='android.permission.WRITE_EXTERNAL_STORAGE'
uses-permission: name='android.permission.VIBRATE'
uses-permission: name='android.permission.MODIFY_AUDIO_SETTINGS'
```

MainActivity.java | proguard-rules.pro | android-normal-debug.apk

com.google.samples.apps.iosched (version 4.3.0d)

Raw File Size: **5.8 MB**, Download Size: **4.9 MB**

Compare with...

File	Raw File Size	Download Size	% of Total Download size
classes2.dex	670.2 KB	254 KB	5%
META-INF	186.5 KB	57.3 KB	1.1%
assets	83.3 KB	26.3 KB	0.5%
AndroidManifest.xml	23.7 KB	4.7 KB	0.1%
NOTICE_firebase_jvm	2 KB	1 KB	0%

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:versionCode="430"
    android:versionName="4.3.0d"
    android:installLocation="0"
    package="com.google.samples.apps.iosched"
    platformBuildVersionCode="23"
    platformBuildVersionName="6.0-2704002">

    <uses-sdk
        android:minSdkVersion="16"
        android:targetSdkVersion="23" />

    <permission
        android:label="@ref/0x7f0a00b8"
        android:name="com.google.samples.apps.iosched.permission.WRITE_SCHEDULE"
        android:protectionLevel="0x0"
```

CAN WE CHANGE STUFF?

```
➜ aapt add -v cool.apk  "assets/thelastjedi.txt"

 ‘assets/thelastjedi.txt'...
```

```
➜ aapt add -v cool.apk  "assets/thelastjedi.txt"

 ‘assets/thelastjedi.txt'...
```

```
➜ aapt list -a cool.apk | grep "assets"
assets/info_en.html
assets/new_service_en.html
assets/style.css
assets/thelastjedi.txt
```

```
➜ aapt remove cool.apk  "assets/thelastjedi.txt"
 ‘assets/thelastjedi.txt'...
```

```
➜ aapt list -a cool.apk | grep "assets"
assets/info_en.html
assets/new_service_en.html
assets/style.css
```

```
➜ adb shell install cool.apk

Failed to install cool.apk: Failure [INSTALL_PARSE_FAILED_NO_CERTIFICATES:
Failed to collect certificates from /data/app/vmdl1746107370.tmp/base.apk:
META-INF/CERT.SF indicates /data/app/vmdl1746107370.tmp/base.apk is signed
using APK Signature Scheme v2, but no such signature was found. Signature
stripped?]
```

```
➜ jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -
keystore valid.keystore -storepass <storepass> cool.apk alias


jar signed.

Warning:
No -tsa or -tsacert is provided and this jar is not timestamped. Without a
timestamp, users may not be able to validate this jar after the signer
certificate's expiration date (yyyy-mm-dd) or after any future revocation date.
```

APKTOOL

- a tool for reverse engineering (3rd party, closed, binary Android apps)
- disassembling resources (resources.arsc, classes.dex, 9.png, XMLs)
- rebuilding decoded resources (APK/JAR)

https://ibotpeaches.github.io/Apktool/

APKTOOL

- a tool for reverse engineering (3rd party, closed, binary Android apps)
- disassembling resources (resources.arsc, classes.dex, 9.png, XMLs)
- rebuilding decoded resources (APK/JAR)

https://ibotpeaches.github.io/Apktool/

APKTOOL

- Java 7+
- download apktool.jar
- wrapper script (or java -jar apktool.jar)
- script and jar to /usr/local/bin
- run **apktool** in terminal

```
➜ apktool
Apktool v2.2.2 - a tool for reengineering Android apk files
with smali v2.1.3 and baksmali v2.1.3
Copyright 2014 Ryszard Wiśniewski <brut.alll@gmail.com>
Updated by Connor Tumbleson <connor.tumbleson@gmail.com>

usage: apktool
 -advance,--advanced   prints advance information.
 -version,--version    prints the version then exits
usage: apktool if|install-framework [options] <framework.apk>
 -p,--frame-path <dir>   Stores framework files into <dir>.
 -t,--tag <tag>          Tag frameworks using <tag>.
usage: apktool d[ecode] [options] <file_apk>
 -f,--force              Force delete destination directory.
 -o,--output <dir>       The name of folder that gets written. Default is apk.out
 -p,--frame-path <dir>   Uses framework files located in <dir>.
 -r,--no-res             Do not decode resources.
 -s,--no-src             Do not decode sources.
 -t,--frame-tag <tag>    Uses framework files tagged by <tag>.
usage: apktool b[uild] [options] <app_path>
 -f,--force-all          Skip changes detection and build all files.
 -o,--output <dir>       The name of apk that gets written. Default is dist/name.apk
 -p,--frame-path <dir>   Uses framework files located in <dir>.

For additional info, see: http://ibotpeaches.github.io/Apktool/
For smali/baksmali info, see: https://github.com/JesusFreke/smali
```

SMALI

- .dex -> smali
- you can learn to read it
- begin with simpler examples

```
➜  apks apktool d -f cool.apk

I: Using Apktool 2.2.2 on cool.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /Users/abaotic/Library/apktool/
framework/1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
➜  apks
```

```
➜ ls cool

AndroidManifest.xml apktool.yml assets  lib
original    res     smali     unknown
```

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    android:installLocation="internalOnly"
    package="abaotic.demo.development"
    platformBuildVersionCode="25"
    platformBuildVersionName="7.1.1">
        <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
        <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
        <uses-permission android:name="android.permission.INTERNET"/>
        <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
    <application
        android:allowBackup="false"
        android:debuggable="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/application_launcher_title"
        android:name="abaotic.demo.DemoApplication"
        android:networkSecurityConfig="@xml/network_security_config">
```

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout android:orientation="vertical"
android:layout_width=“match_parent"
android:layout_height="match_parent"
  xmlns:android="http://schemas.android.com/apk/res/android">
    <TextView android:textSize="18.0dip" android:textStyle="bold"
android:id="@id/title" android:background="@android:color/transparent"
android:padding="10.0dip" android:layout_width="match_parent"
android:layout_height="wrap_content" android:text="@string/news"
android:layout_weight="0.0" />
    <WebView android:id="@id/webView"
android:layout_width="match_parent" android:layout_height="0.0dip"
android:layout_weight="1.0" />
</LinearLayout>
```

```
➜  cool apktool b .

I: Using Apktool 2.2.2
I: Checking whether sources has changed...
I: Smaling smali folder into classes.dex...
I: Checking whether resources has changed...
I: Building resources...
I: Copying libs... (/lib)
I: Building apk file...
I: Copying unknown files/dir...
```

```
➜  dist adb install cool.apk

Failed to install cool.apk: Failure
[INSTALL_PARSE_FAILED_NO_CERTIFICATES: Failed to collect certificates
from /data/app/vmdl904970069.tmp/base.apk: Attempt to get length of null
array]
```

```
➜ dist jarsigner -verbose -sigalg SHA1withRSA
-digestalg SHA1 -keystore production_keystore
-storepass <storepass> cool.apk <alias>

adding: META-INF/MANIFEST.MF
adding: META-INF/ANDROID.SF
adding: META-INF/ANDROID.RSA
signing: AndroidManifest.xml
…
➜ dist adb install cool.apk
Success
```

USE?

- simple
- easy to use
- might not work (missing proper framework files)

IS THIS A PROBLEM?

com.google.samples.apps.iosched-333.apk

Classes | Methods count

- com.google.samples.apps
  - AndroidManifest.xml
  - classes
  - res

www.classyshark.com

http://www.retrojunkie.com/asciiart/animals/sharks.htm

ClassyShark ver.6.5 powered by SilverGhost

CLASSYSHARK

- https://github.com/google/android-classyshark
- browse components
- inspect method count
- export a report with all relevant info


ANDROGUARD
ALWAYS READY ALWAYS THERE

ANDROGUARD

- https://github.com/androguard/androguard
- written in Python
- advanced, many options
- reverse engineering, malware analysis

```
python androlyze.py -s
Androlyze version 3.0
In [1]: a, d, dx = AnalyzeAPK(“cool.apk")

In [2]: a.get_main_activity()
Out [2]: u’abaotic.demo.development.CoolActivity’

a.get_permissions()
a.get_services()
a.get_receivers()
```

CHARLES

- HTTP proxy / HTTP monitor / reverse proxy
- view all traffic between the Client and Internet
- 30 day trial

FEATURES

- SSL proxying (MITM)

```
<
   android:name="AppName"
   …

   android:networkSecurityConfig=
      "@xml/network_security_config">
```

```
<network-security-config>
    <base-config>
        <trust-anchors>
            <certificates src="system" />
            <certificates src="user" />
        </trust-anchors>
    </base-config>
</network-security-config>
```

FEATURES

- bandwidth throttling
- **repeat** requests
- **intercept** and **edit** requests or responses

FEATURES

- bandwidth throttling
- **repeat** requests
- **intercept** and **edit** requests or responses

```
{
    "first_name": "Leia",
    "last_name": "Organa",
    "is_jedi": true
}
```

```
{
    "first_name": "Leia",
    "last_name": "Organa",
    "is_jedi": false
}
```

TOOLS - CHARLES EDIT REQUEST/RESPONSE

```
{
    [ ]
}
```

```
{
    "username": "ab39079",
    "password": "nicetrybutnocigar",
    "serial_number": "1234567890",
    "keep_data": true
}
```

WHY PEN TEST?

- analyse to learn and save time (learn by examples)
- detect leaks and common oversights
- proof your app (tampering detection)
- fallback commercial solutions

@abaotic

THANK YOU

REFERENCES

- Devknox
  https://devknox.io/
- Google Play pre-launch report
  https://support.google.com/googleplay/android-developer/answer/7002270?hl=en
- Penetration testing
  https://en.wikipedia.org/wiki/Penetration_test
- ADB shell commands
  http://adbshell.com/commands
- APK Analyzer
  https://developer.android.com/studio/build/apk-analyzer.html

REFERENCES

- Apktool
  https://ibotpeaches.github.io/Apktool/
- ClassyShark
  https://github.com/google/android-classyshark
- Charles
  https://www.charlesproxy.com/
- Androguard
  https://github.com/androguard/androguard
- **https://github.com/OWASP/owasp-mstg**